UC-7110 Series User's Manual

www.moxa.com/product

First Edition, September 2004

Moxa Technologies Co., Ltd.
Tel: +886-2-8919-1230
Fax: +886-2-8919-1231
www.moxa.com
support@moxa.com.tw (Worldwide)
support@moxa.com (The Americas)

UC-7110 Series User's Manual

The software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

Copyright Notice

Copyright © 2004 Moxa Technologies Co., Ltd.
All rights reserved.
Reproduction without permission is prohibited.

Trademarks

MOXA is a registered trademark of The Moxa Group.
All other trademarks or registered marks in this manual belong to their respective manufacturers.

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.

Moxa provides this document "as is," without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Moxa reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Moxa assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Table of Contents

Chapter 1 Introduction 1-1
Overview 1-2
Package Checklist 1-2
Product Features 1-2
Product Specifications 1-3
Hardware Specifications (UC-7110-LX) 1-3
Software Specifications (UC-7110-LX) 1-4
Hardware Block Diagram 1-5
Appearance 1-5
Dimensions 1-6
Installing UC-7110 1-7
LED Indicators 1-8
Wiring Requirements 1-8
Connecting the Power 1-9
Grounding UC-7110 1-9
Connecting Data Transmission Cables 1-9
Connecting to the Network 1-9
Connecting to a Serial Device 1-10
Serial Console Port 1-10
Additional Functions 1-11
Reset Button 1-11
Real Time Clock 1-12

Chapter 2 Getting Started 2-1
Powering on UC-7110 2-2
Connecting UC-7110 to a PC 2-2
Console Port 2-2
Telnet 2-3
Configuring the Ethernet Interface 2-4
Developing Your Applications 2-6
Installing the UC-7110 Tool Chain 2-6
Compiling Hello.c 2-7
Uploading “Hello” to UC-7110 2-8
Running “Hello” on UC-7110 2-9
Make File Example Code 2-10

Chapter 3 Software Package 3-1
UC-7110 Software Architecture 3-2
Read-Only File System (ROMFS) 3-3
Journaling Flash File System (JFFS2) 3-3
UC-7110 Software Package 3-4

Chapter 4 Configuring UC-7110 4-1
Enabling and Disabling Daemons 4-2
Adding a Web Page 4-3
Configuring Dial-in/Dial-out Service 4-4
Dial-out Service 4-4
Dial-in Service 4-4
How to Mount a Remote NFS Server 4-5

Upgrading the Kernel & Root File System 4-5
Upgrading the User Directory 4-7
User Directory Backup—UC-7110 to PC 4-8
Loading Factory Defaults 4-9
Mirroring the Application Program and Configuration 4-9
How to Autostart User Applications on Bootup 4-9
Checking the Kernel and User Directory Versions 4-9

Chapter 5 UC-7110 Device API 5-1
RTC (Real Time Clock) 5-2
Buzzer 5-2
UART Interface 5-2

Chapter 6 UC Finder 6-1
Windows UC Finder 6-2
Linux ucfinder 6-5

Appendix A System Commands A-1
busybox (V0.60.4): μClinux normal command utility collection A-1
File manager A-1
Editor A-1
Network A-2
Process A-2
Other A-2
MOXA Special Utilities A-2

Appendix B SNMP Agent with MIB II & RS-232 Like Group B-1

Appendix C FAQ for UC-7110 ..... C-1

Appendix D Service Information D-1
MOXA Internet Services D-2
Problem Report Form D-3
Product Return Procedure D-4

1

Introduction

The MOXA UC-7110 Series of RISC-based Communication Platforms features two RS-232/422/485 serial ports, and dual 10/100 Mbps Ethernet ports, making these products ideal for embedded OS applications.

This chapter covers the following topics:

- **Overview**
- **Package Checklist**
- **Product Features**
- **Product Specifications**
- **Hardware Block Diagram**
- **Appearance**
- **Dimensions**
- **Installing UC-7110**
- **LED Indicators**
- **Wiring Requirements**
  - Connecting the Power
  - Grounding UC-7110
- **Connecting Data Transmission Cables**
  - Connecting to the Network
  - Connecting to a Serial Device
  - Serial Console Port
- **Additional Functions**
  - Reset Button (not yet activated)
  - Real Time Clock

Overview

The UC-7110 Series of RISC-based Communication Platforms are ideal for your embedded applications. UC-7110 comes with two RS-232/422/485 serial ports and dual 10/100 Mbps Ethernet LAN ports to provide users with a versatile communication platform.

UC-7110 uses the ARM9 166 Mhz RISC CPU. Unlike the X86 CPU, which uses a CISC design, the ARM9's RISC design architecture and modern semiconductor technology provide UC-7110 with a powerful computing engine and communication functions, but without generating too much heat. The built-in 8 MB NOR Flash ROM and 16 MB SDRAM give you enough storage capacity to run applications on UC-7110. The dual LAN ports built into the ARM9 make UC-7110 an ideal communication platform for some of the more simple data acquisition and protocol conversion applications, and the two RS-232/422/485 serial ports allow you to connect a variety of different types of serial device.

The pre-installed µClinux operating system provides an open software operating system for software program development. This means that software written for desktop PCs is easily ported to UC-7110 by using a GNU cross complier, so that you will not need to spend time modifying existing software code. The Operating System, device drivers, and your own software can all be stored in UC-7110's Flash memory.

Package Checklist

The UC-7110 Series model currently available is:

UC-7110-LX RISC-based Universal Communicator with 2 Serial Ports, Dual Ethernet, µClinux OS

UC-7110 Series products are shipped with the following items:

- 1 UC-7110
- UC-7110 Quick Installation Guide
- Universal Communicator Documentation & Software CD-ROM
- Ethernet cross-over cable: RJ45 to RJ45, 100 cm
- Console port cable
  CBL-4PINDB9F-100: 4-pin header to DB9 (Female) cable, 100 cm
- Power Adaptor
- Product Warranty Booklet

NOTE: *Notify your sales representative if any of the above items is missing or damaged.*

Product Features

UC-7110 Series products have the following features:

- ARM9 166 MHz Processor
- On-board 16 MB RAM, 8 MB Flash ROM
- Two RS-232/422/485 serial ports
- Dual 10/100 Mbps Ethernet
- µClinux-Ready communication platform
- DIN-Rail or wall mounting installation
- Robust fanless design

Product Specifications

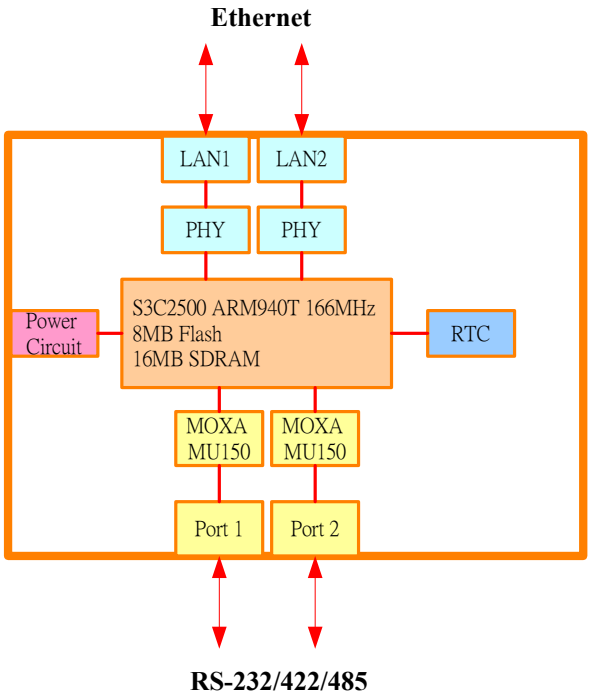
Hardware Specifications (UC-7110-LX)

CPU	ARM9-based 32-bit RISC CPU, 166 Mhz
RAM	16 MB (12 MB of user programmable space)
Flash	8 MB (4 MB of user programmable space)
LAN	Auto-sensing 10/100 Mbps x 2
LAN Protection	Built-in 1.5 KV magnetic isolation
Serial Ports	The two RS-232/422/485 ports support: RS-232 signals: TxD, RxD, DTR, DSR, RTS, CTS, DCD, GND RS-422 signals: TxD+, TxD-, RxD+, RxD-, GND 4-wire RS-485 signals: TxD+, TxD-, RxD+, RxD-, GND 2-wire RS-485 signals: Data+, Data-, GND
Serial Protection	15 KV ESD for all signals
Data bits	5, 6, 7, 8
Stop bits	1, 1.5, 2
Parity	None, even, odd, space, mark
Flow Control	RTC/CTS, XON/XOFF
Speed	50 bps to 921.6 Kbps
Real Time Clock	Yes
Buzzer	Yes
Console Port	RS-232, 3-wire (Tx, Rx, GND) (19200, n, 8 , 1)
LEDs	Ready Serial Tx, Rx (2 of each) LAN 10/100 (one on each LAN connector)
Gross Weight	190 g
Power input	12-48 VDC
Power Consumption	290 mA @ 12 VDC
Operating temperature	-10 to 60℃, (14 to 140F), 5 to 95%RH
Storage temperature	-20 to80℃, (-4 to 185F), 5 to 95%RH
Regulatory Approvals	EMC: FCC Class A, CE Class A Safety: UL, CUL, TÜV
Warranty	5 years

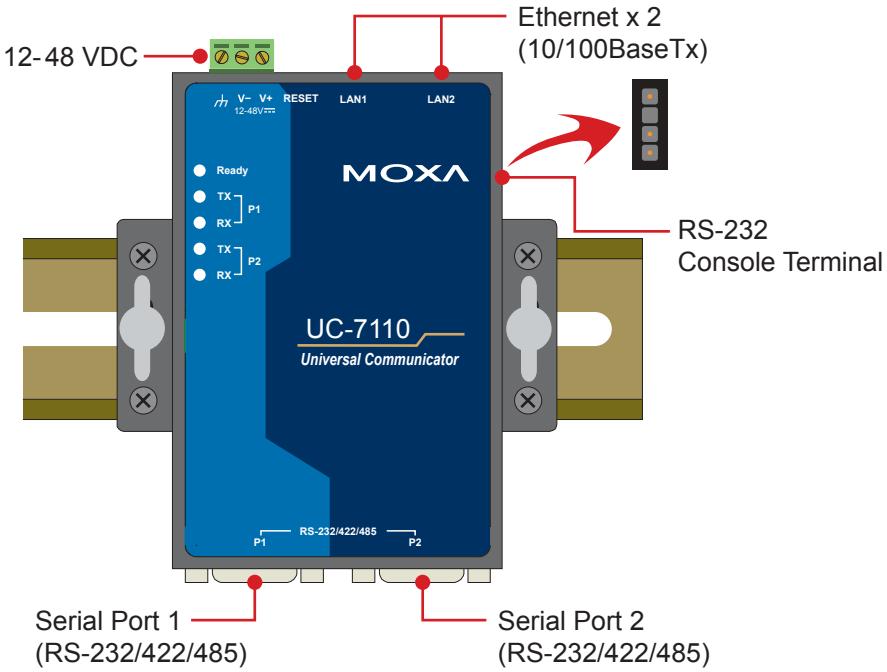
Software Specifications (UC-7110-LX)

Kernel	μClinux Kernel 2.4.22
Protocol Stack	ARP, ICMP, IPV4, TCP, UDP, FTP, Telnet, NTP, SNMP V1, HTTP, PPP, CHAP, PAP, NFS
JFFS2 File System	User Directory (Read / Write)
ROMFS	Kernel, Root File System (Read Only)
Sash (V1.1.1)	Simple OS shell command
Pppd (V1.13)	dial in/out over serial port daemon
snmpd	SNMP V1 Agent daemon
busybox (V0.60.4)	Linux normal command utility
Tinylogin (V0.80)	login and user manager utility
Telnetd (V1.9)	Telnet server daemon
telnet (busybox V0.60.4)	Telnet client program
inetd (Distributed 20030522)	TCP server manager program
ftpd (V8.5)	FTP server program
ftp (V0.16)	FTP client program
boa	Web server daemon
ntpdate (V4.1.1)	Network Time Protocol client utility
Tool Chain	
Arm-elf-gcc (V2.95.3)	C/C++ PC Cross Compiler
uClib (V0.9.19)	POSIX standard library
UC Finder	UC's LAN IP broadcast searching utility for Windows and Linux

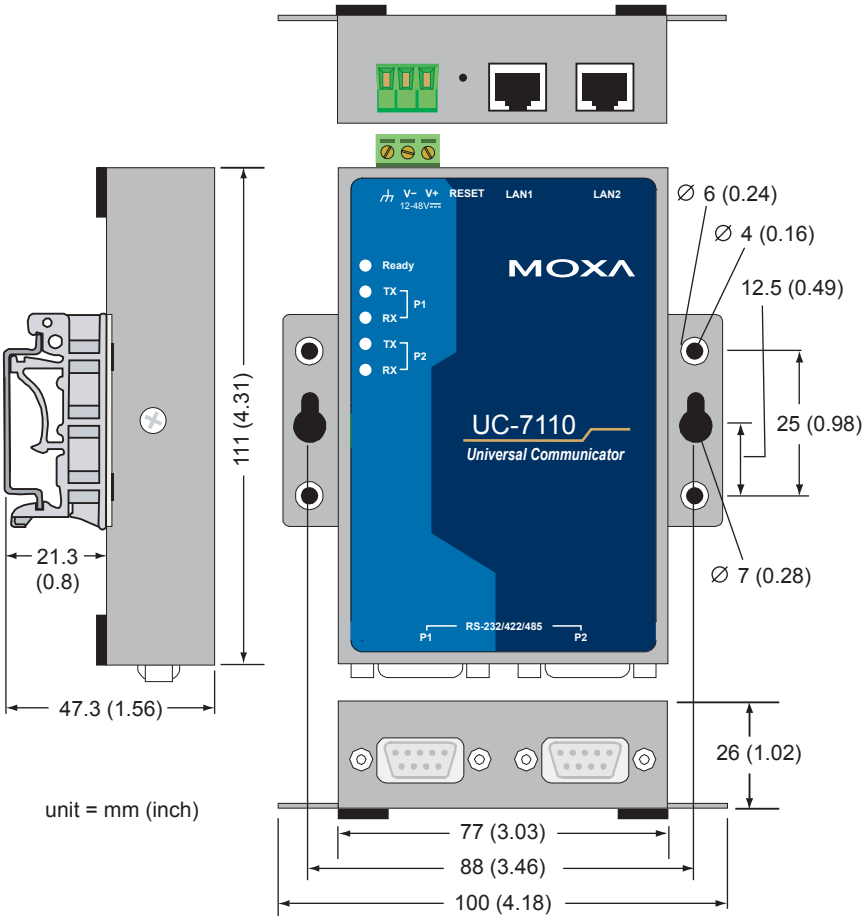
Hardware Block Diagram



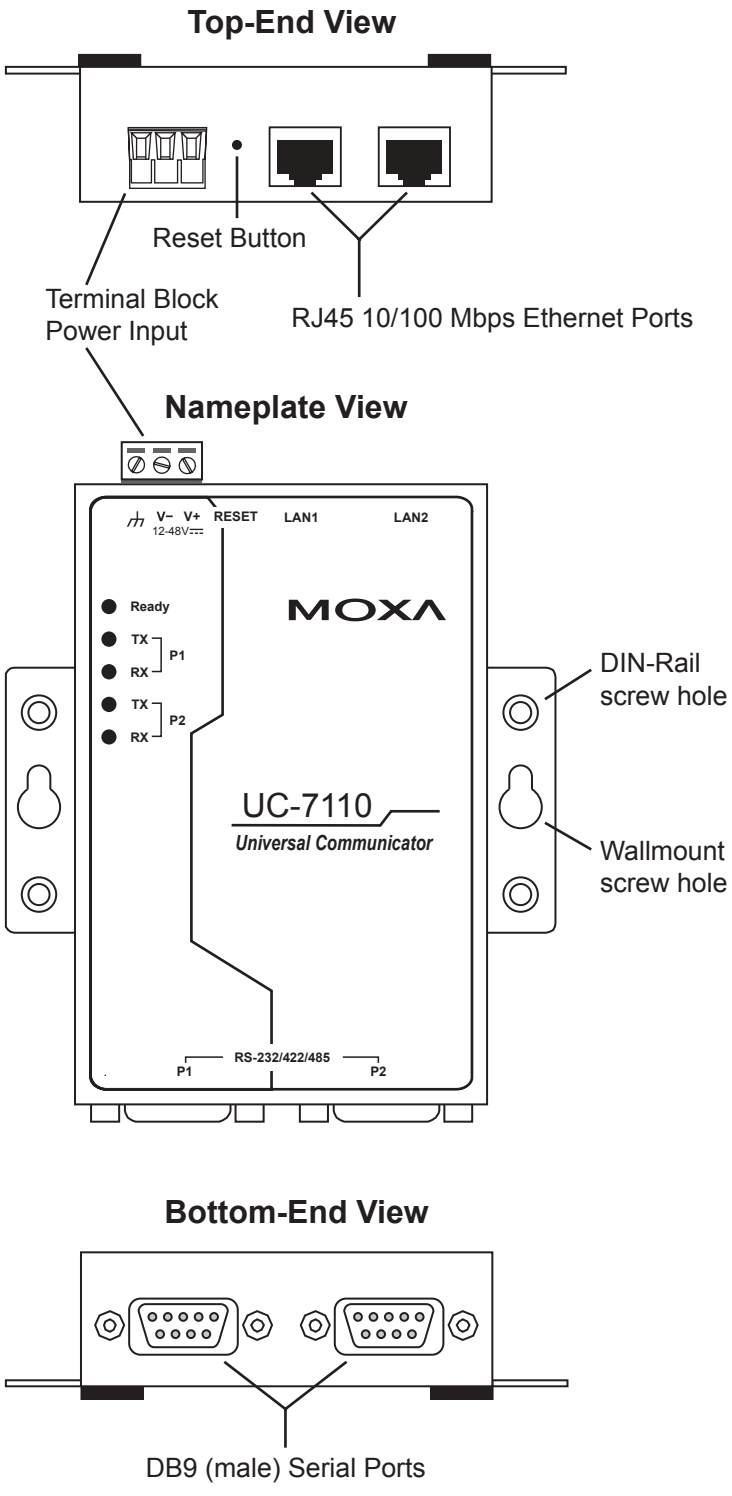
Appearance



Dimensions


V- V+
12-48V
RESET
LAN1
LAN2
Ready
TX
RX
P1
TX
RX
P2
MOXA
UC-7110
Universal Communicator
RS-232/422/485
P1
P2
∅ 6 (0.24)
∅ 4 (0.16)
12.5 (0.49)
25 (0.98)
∅ 7 (0.28)
111 (4.31)
21.3
(0.8)
47.3 (1.56)
26 (1.02)
77 (3.03)
88 (3.46)
100 (4.18)
unit = mm (inch)

Installing UC-7110



LED Indicators

The following table explains the function of the five LED indicators located on UC-7110's top panel.

LED Name	LED Color	LED Function
Ready	Green	Power is on and functioning normally.
P1/P2 (Tx)	Green	Serial port 1/2 is transmitting data.
	Off	Serial port 1/2 is not transmitting data.
P1/P2 (Rx)	Yellow	Serial port 1/2 is receiving data.
	Off	Serial port 1/2 is not receiving data.

Wiring Requirements

This section describes how to connect UC-7110 to serial devices.

You should heed the following common safety precautions before proceeding with the installation of any electronic device:

- Use separate paths to route wiring for power and devices. If power wiring and device wiring paths must cross, make sure the wires are perpendicular at the intersection point.

  NOTE: Do not run signal or communication wiring and power wiring in the same wire conduit. To avoid interference, wires with different signal characteristics should be routed separately.

- Use the type of signal transmitted through a wire to determine which wires should be kept separate. The rule of thumb is that wiring that shares similar electrical characteristics can be bundled together.
- Keep input wiring and output wiring separate.
- It is advisable to label the wiring to all devices in the system.

ATTENTION

Safety First!

Be sure to disconnect the power cord before installing and/or wiring your UC-7110.

Wiring Caution!

Calculate the maximum possible current in each power wire and common wire. Observe all electrical codes dictating the maximum current allowable for each wire size.

If the current goes above the maximum ratings, the wiring could overheat, causing serious damage to your equipment.

Temperature Caution!

Be careful when handling UC-7110. When plugged in, UC-7110's internal components generate heat, and consequently the outer casing may feel hot to the touch.

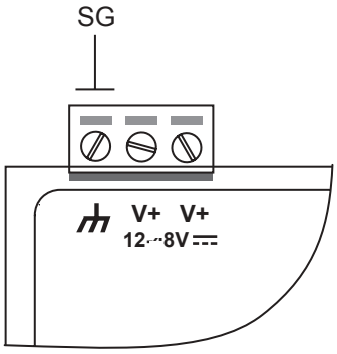
Connecting the Power

Connect the "live-wire" end of the 12-48 VDC power adaptor to UC-7110's terminal block. If the power is properly supplied, the "Ready" LED will show a solid green color after a 25 to 30 second delay.

Grounding UC-7110

Grounding and wire routing helps limit the effects of noise due to electromagnetic interference (EMI). Run the ground wire from the ground screw to the grounding surface prior to connecting devices.

ATTENTION This product should be mounted to a well-grounded mounting surface such as a metal panel.



SG: The *Shielded Ground* (sometimes called *Protected Ground*) contact is the left most contact of the 3-pin power terminal block connector when viewed from the angle shown here. Connect the SG wire to an appropriate grounded metal surface.

Connecting Data Transmission Cables

This section describes how to connect UC-7110 to the network, serial devices, and serial COM terminal.

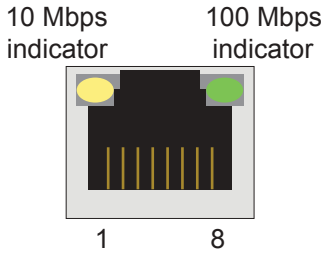
Connecting to the Network

Connect one end of the Ethernet cable to UC-7110's 10/100M Ethernet port and the other end of the cable to the Ethernet network. If the cable is properly connected, UC-7110 will indicate a valid connection to the Ethernet in the following ways:

- The top-right LED on the connector maintains a solid green color when connected to a 100 Mbps Ethernet network.
- The top-left LED on the connector maintains a solid orange color when connected to a 10 Mbps Ethernet network.
- The LEDs will flash when Ethernet packets are being transmitted or received.

The 10/100 Mbps Ethernet LAN 1 and LAN 2 ports use 8-pin RJ45 connectors. Pinouts for these ports are given in the following diagram.

8-pin RJ45



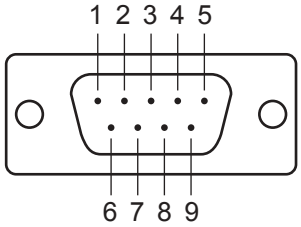
Pin	Signal
1	ETx+
2	ETx-
3	ERx+
4	---
5	---
6	ERx-
7	---
8	---

Connecting to a Serial Device

Connect the serial cable between UC-7110 and the serial device(s).

Serial ports P1 and P2 use male DB9 connectors, and can be configured for RS-232/422/485 by software. The pin assignments are shown in the following table:

DB9 Male Port



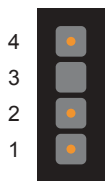
RS-232/422/485 Pinouts

Pin	RS-232	RS-422	RS-485 (4-wire)	RS-485 (2-wire)
1	DCD	TxDA(-)	TxDA(-)	---
2	RxD	TxDB(+)	TxDB(+)	---
3	TxD	RxDB(+)	RxDB(+)	DataB(+)
4	DTR	RxDA(-)	RxDA(-)	DataA(-)
5	GND	GND	GND	GND
6	DSR	---	---	---
7	RTS	---	---	---
8	CTS	---	---	---

Serial Console Port

The serial console port is a 4-pin pin-header RS-232 port. It is designed for serial console terminals, which are useful for identifying the UC-7110 boot up message.

Serial Console Port & Pinouts



Pin	Signal
1	TxD
2	RxD
3	NC
4	GND

Serial Console Cable

Additional Functions

Reset Button

Press the "RESET" button continuously for more than 5 seconds to load the factory default configuration. After loading the factory default, the system will reboot automatically. The System Ready LED will be blinking for the first 5 seconds. We recommend that you only use this function if the software is not working properly. To reset the µClinux system software, always use the software reboot command />reboot to protect the integrity of data in the process of being transmitted. The reset button is not designed to Hard Reboot UC-7110.

ATTENTION

Resetting to factory default will not format the user directory and erase all of the user's data. Loading factory default will only load the configuration file. The files in UC-7110 that will be changed include:

a. /etc/boa.conf
b. /etc/hosts
c. /etc/inittab
d. /etc/password
e. /etc/ramfs.img
f. /etc/resolv.conf
g. /etc/version
h. /etc/group
i. /etc/inetd.conf
j. /etc/motd
k. /etc/protocols
l. /etc/rc
m. /etc/services
o. /home/httpd/boa.conf
p. /home/httpd/index.html

ATTENTION

This function only takes effect when the user directory is working correctly. If the user directory has crashed, the kernel will automatically load factory default.

Real Time Clock

UC-7110's real time clock is powered by a lithium battery. We strongly recommend that you do not replace the lithium battery without the help of Moxa's support team. If the battery needs to be changed, contact the Moxa RMA service team for RMA service.

ATTENTION

The battery may explode if replaced by the incorrect type. To avoid this potential danger, always be sure to use the correct type of battery.

2

Getting Started

In this chapter, we explain the basic procedure for getting UC-7110 connected.

This chapter covers the following topics:

- **Powering on UC-7110**
- **Connecting UC-7110 to a PC**
- **Configuring the Ethernet Interface**
- **Developing Your Applications**
  - Installing the UC-7110 Tool Chain
  - Compiling Hello.c
  - Uploading “Hello” to UC-7110
  - Running “Hello” on UC-7110
  - Make File Sample Code

Powering on UC-7110

Connect the SG wire to the Shielded Contact located on the upper left corner of the UC-7110, and then power on UC-7110 by connecting the power adaptor. It takes about 16 seconds for the system to boot up. Once the system is ready, the Ready LED will light up.

ATTENTION

After connecting UC-7110 to the power supply, it will take about 16 seconds for the operating system to boot up. The green Ready LED will not turn on until the operating system is ready.

Connecting UC-7110 to a PC

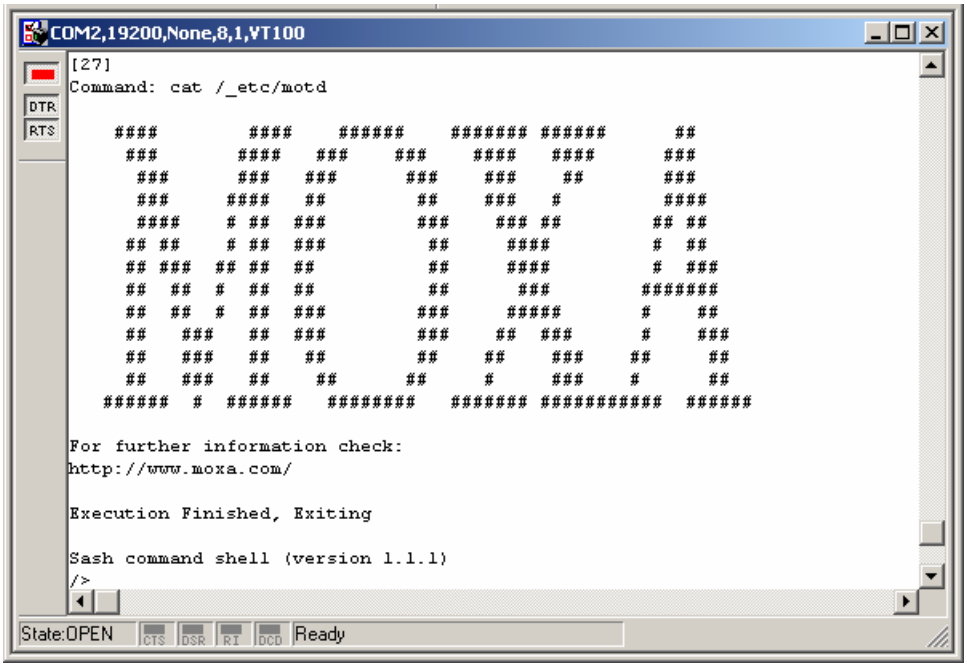
There are two ways to connect UC-7110 to a PC.

Console Port

The serial console port offers users with a convenient mean of connecting to UC-7110. This method is particularly useful when using UC-7110 for first time. Since the communication is over a direct serial connection, you do not need to know either of the IP addresses in order to make contact.

Use serial console port settings shown at the right. Once the connection is established, the following window will open.

Serial Console Port Settings	
Baud rate	19200 bps
Parity	None
Data bits	8
Stop bits	1
Flow Control	None
Terminal	VT100



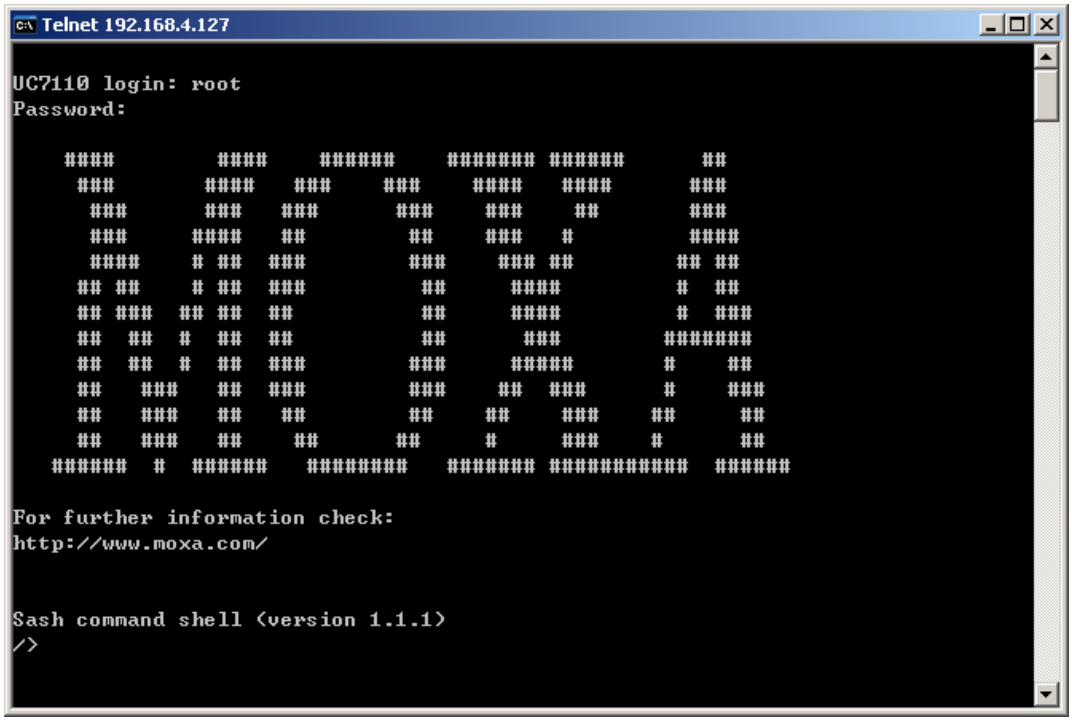
Telnet

If you know at least one of the two IP addresses and netmasks, then you can use Telnet to connect to UC-7110's console.

	Default IP Address	Default Netmask
LAN 1	192.168.3.127	255.255.255.0
LAN 2	192.168.4.127	255.255.255.0

Telnet can be used locally by using a cross-over Ethernet cable to connect your computer to UC-7110, or over a LAN or the Internet. The default IP addresses and netmasks are shown above. To login, type the Login name and password as requested. The defaults are:

Login: root
Password: root



Once you reach the "Sash command shell" you can proceed to configure UC-7110's network settings, as described in the next section.

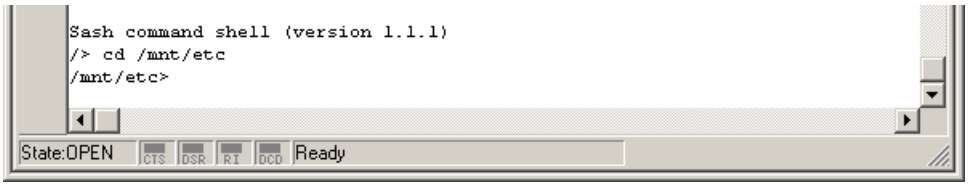
ATTENTION

- **Serial Console Reminder**: Remember to choose **VT100** as the terminal type. Use the CBL-RJ45F9-150 cable that comes with UC-7110 to connect to the serial console port. If you do not get connected on the first try, unplug and then re-plug UC-7110's power cord.
- **Telnet Reminder**: When connecting to UC-7110 over a LAN, you must configure your PC's Ethernet card to be on the same subnet as the UC-7110 you wish to contact.

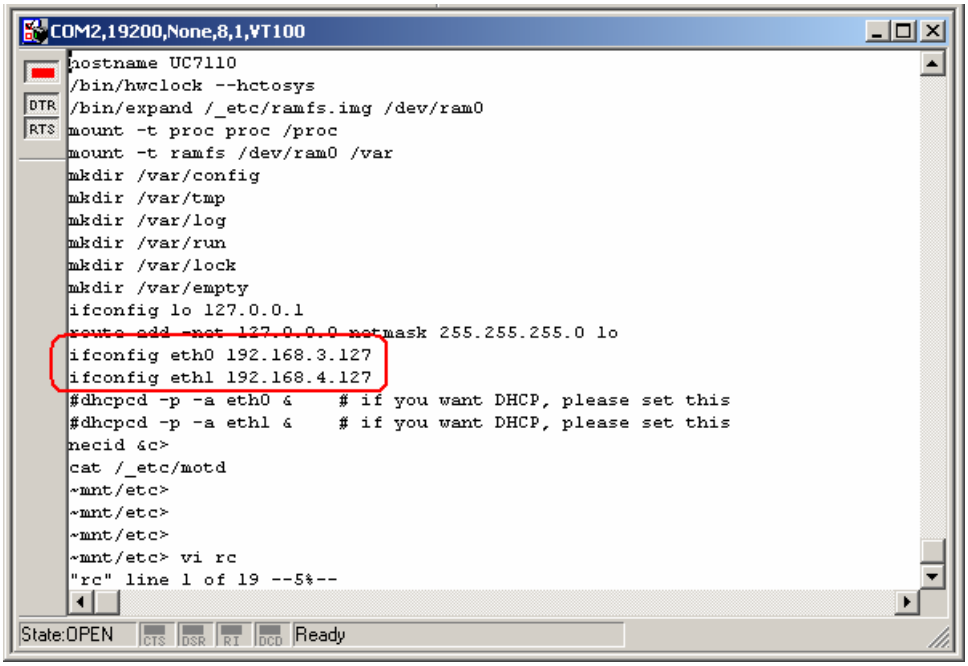
Configuring the Ethernet Interface

In this section we use the serial console to explain how to modify UC-7110's network settings.

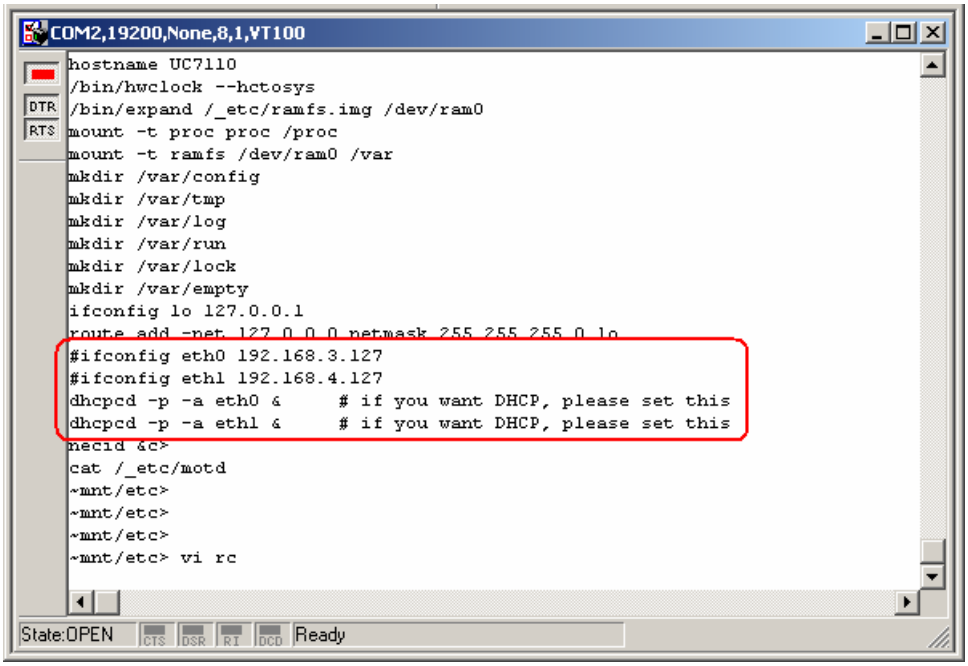
1. Change directories by issuing the command **`cd /mnt/etc`**.



2. Type the command **`vi rc`** to use VI Editor to edit the configuration file. The IP addresses for UC-7110's LAN1 and LAN2 are given as
`ifconfig eth0 192.168.3.127`
`ifconfig eth1 192.168.4.127`
as shown in the following figure. Edit these two lines to modify the static IP addresses.



3. You may also configure UC-7110 to request IP addresses from a DHCP server. In this case, use the sharp sign (#) to comment out one or both "ifconfig" lines, and then delete the sharp sign in front of the "dhcpcd" lines, as shown below. Note that UC-7110 will send out DHCP broadcast packets, and then get the IP addresses from the first DHCP server that responds.



4. Issue the vi "write" command to save the file, and then reboot. Since UC-7110 only reads the "rc" file when booting up, you must reboot (e.g., by issuing the vi **`reboot`** command) for the changes to take affect.

ATTENTION

You may reset the IP address immediately by issuing the command

`ifconfig eth0 192.168.5.127`

(This will change the IP address of LAN1.) Issuing this command will NOT, however, update the "rc" file in UC-7110's flash memory, so the next time you reboot, the IP address will revert to its previous value.

Developing Your Applications

Step 1:
Connect UC-7110 to a Linux PC.

Step 2:
Install Tool Chain (GNU Cross Compiler & uClibc).

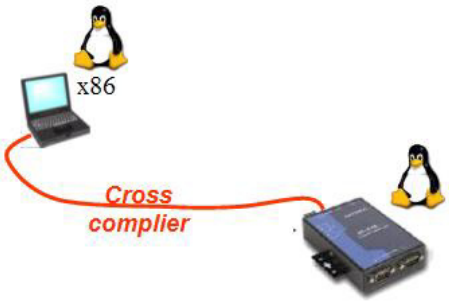
Step 3:
Configure cross compiler and uClibc environment variables.

Step 4:
Code & compile your program.

Step 5:
Download program to UC-7110 via FTP or NFS.

Step 6:
Debug the program. If the program is OK, proceed to Step 7. If the program needs to be modified, go back to Step 4.

Step 7:
Back up the user directory, and distribute the code to additional UC-7110 units.



Installing the UC-7110 Tool Chain

The PC must have the Linux Operating System pre-installed to install the UC-7110 GNU Tool Chain. Redhat 7.3/8.0 and compatible versions are recommended. The Tool Chain requires about 100 MB of hard disk space (on your PC). The UC-7110 Tool Chain can be found on the UC-7110 CD. To install Tool Chain, insert the CD into your PC and then issue the following command:

```
#mount /dev/cdrom /mnt/cdrom
```

Next, run the following script from the root to installs the compilers, linkers, and libraries in the **/usr/local** directory:

```
#sh /mnt/cdrom/tool-chain/arm-elf-tools-20030314.sh
```

The Tool Chain installation will take a few minutes to complete.

ATTENTION

You can download the Tool Chain from Moxa's website (go to the UC-7110 product page, click on the Documentation & Drivers link, and then click on **Go** under Driver & Software Downloads.

The Tool Chain is also available on the www.uClinux.org website at the following link:

http://www.uClinux.org/pub/uClinux/m68k-elf-tools/arm-elf-tools-20030314.sh.

Compiling Hello.c

The Tool Chain path is:

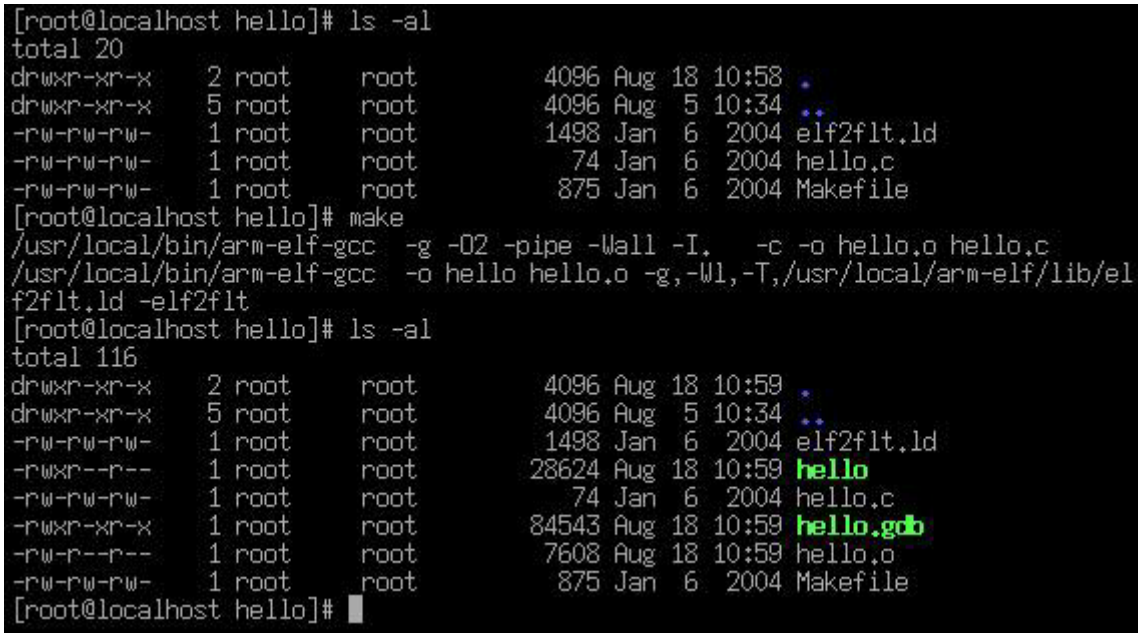
PATH=/usr/local/arm-elf/bin:$PATH

The UC-7110 CD includes several example programs. We use **Hello.c** to illustrate how to compile and run applications.

Issue the following commands from your PC to compile Hello.c:

```
# cd /tmp/
# mkdir example
# cp –r /mnt/cdrom/example/* /tmp/example
```

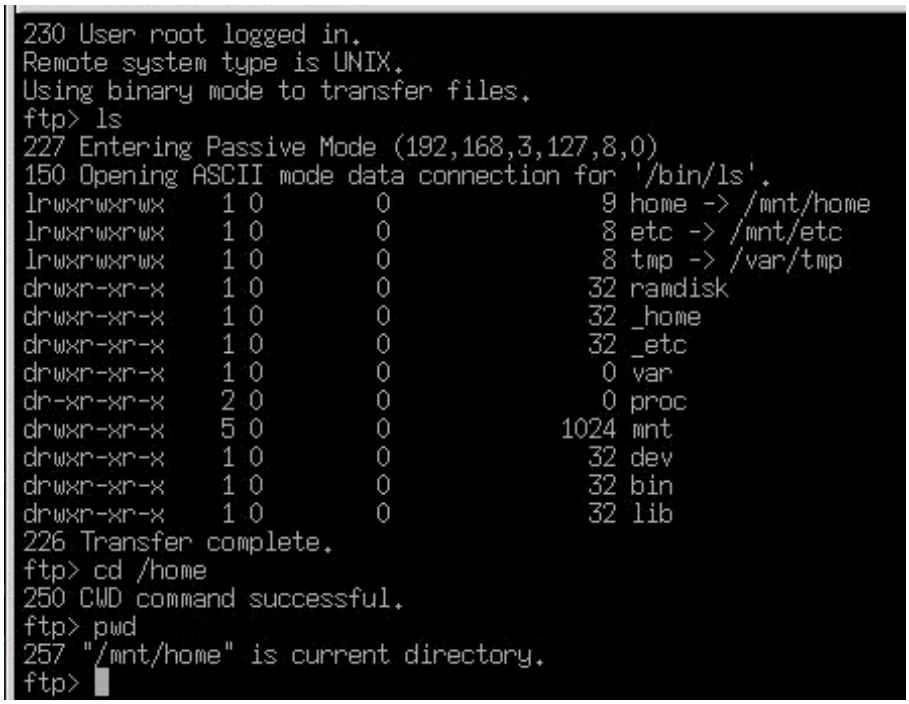
Go to the **Hello** subdirectory, and then issue the command
#make
to compile **Hello.c**. Finally, execute the program to generate **hello** and **hello.gdb**.


[root@localhost hello]# ls -al
total 20
drwxr-xr-x 2 root root 4096 Aug 18 10:58 .
drwxr-xr-x 5 root root 4096 Aug 5 10:34 ..
-rw-rw-rw- 1 root root 1498 Jan 6 2004 elf2flt.ld
-rw-rw-rw- 1 root root 74 Jan 6 2004 hello.c
-rw-rw-rw- 1 root root 875 Jan 6 2004 Makefile
[root@localhost hello]# make
/usr/local/bin/arm-elf-gcc -g -O2 -pipe -Wall -I. -c -o hello.o hello.c
/usr/local/bin/arm-elf-gcc -o hello hello.o -g,-Wl,-T,/usr/local/arm-elf/lib/el
f2flt.ld -elf2flt
[root@localhost hello]# ls -al
total 116
drwxr-xr-x 2 root root 4096 Aug 18 10:59 .
drwxr-xr-x 5 root root 4096 Aug 5 10:34 ..
-rw-rw-rw- 1 root root 1498 Jan 6 2004 elf2flt.ld
-rwxr--r-- 1 root root 28624 Aug 18 10:59 hello
-rw-rw-rw- 1 root root 74 Jan 6 2004 hello.c
-rwxr-xr-x 1 root root 84543 Aug 18 10:59 hello.gdb
-rw-r--r-- 1 root root 7608 Aug 18 10:59 hello.o
-rw-rw-rw- 1 root root 875 Jan 6 2004 Makefile
[root@localhost hello]#

Uploading "Hello" to UC-7110

To use FTP to upload **hello** to UC-7110, issue the following commands from the PC side:

```
#ftp 192.168.3.127
ftp> cd /home
ftp> bin
ftp> put ./hello
ftp> quit
#telnet 192.168.3.127
```


230 User root logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (192,168,3,127,8,0)
150 Opening ASCII mode data connection for '/bin/ls'.
lrwxrwxrwx 1 0 0 9 home -> /mnt/home
lrwxrwxrwx 1 0 0 8 etc -> /mnt/etc
lrwxrwxrwx 1 0 0 8 tmp -> /var/tmp
drwxr-xr-x 1 0 0 32 ramdisk
drwxr-xr-x 1 0 0 32 _home
drwxr-xr-x 1 0 0 32 _etc
drwxr-xr-x 1 0 0 0 var
dr-xr-xr-x 2 0 0 0 proc
drwxr-xr-x 5 0 0 1024 mnt
drwxr-xr-x 1 0 0 32 dev
drwxr-xr-x 1 0 0 32 bin
drwxr-xr-x 1 0 0 32 lib
226 Transfer complete.
ftp> cd /home
250 CWD command successful.
ftp> pwd
257 "/mnt/home" is current directory.
ftp>

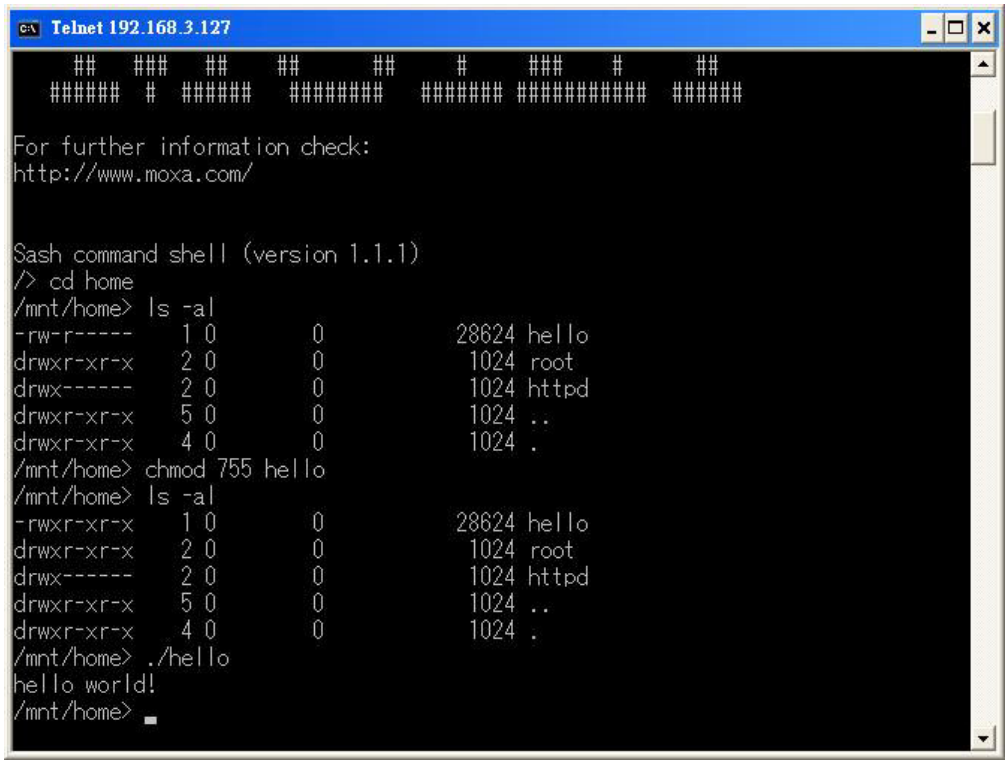
Running "Hello" on UC-7110

To run the "Hello" program issue the following commands from the UC-7110 side:

 # chmod 755 hello

#./hello

The words "Hello World" will be printed on the screen.


Telnet 192.168.3.127
For further information check:
http://www.moxa.com/
Sash command shell (version 1.1.1)
/> cd home
/mnt/home> ls -al
-rw-r----- 1 0 0 28624 hello
drwxr-xr-x 2 0 0 1024 root
drwx------ 2 0 0 1024 httpd
drwxr-xr-x 5 0 0 1024 ..
drwxr-xr-x 4 0 0 1024 .
/mnt/home> chmod 755 hello
/mnt/home> ls -al
-rwxr-xr-x 1 0 0 28624 hello
drwxr-xr-x 2 0 0 1024 root
drwx------ 2 0 0 1024 httpd
drwxr-xr-x 5 0 0 1024 ..
drwxr-xr-x 4 0 0 1024 .
/mnt/home> ./hello
hello world!
/mnt/home>

ATTENTION

Be sure to calculate the amount of Flash Memory used by the User File System in the Flash ROM. Use the following command to determine the amount of memory being used:

/>df -h

```
/> df -h
Filesystem                Size      Used Available Use% Mounted on
rootfs                    1.1M      1.1M         0 100% /
/dev/rom0                 1.1M      1.1M         0 100% /
/dev/mtdblock2            4.0M    652.0k      3.4M  16% /mnt
/>
```

If the flash memory is full, you will no longer be able to save data in Flash ROM. To free up some memory, use the console cable to connect to UC-7110's serial console terminal, and then delete files from the Flash ROM.

Make File Example Code

The following Makefile file example codes are copied from the Hello example from UC-7110's CD-ROM.

```
# Generated automatically from Makefile.in by configure.
# $Id: Makefile.in,v 1.59 2002/03/24 22:20:19 jnelson Exp $

.SUFFIXES:
.SUFFIXES: .o .c
.PHONY: clean mrclean distclean depend all dist

GCC_FLAGS = -Wstrict-prototypes -Wpointer-arith -Wcast-align
-Wcast-qual\
  -Wtraditional\
  -Wshadow\
  -Wconversion\
  -Waggregate-return\
  -Wmissing-prototypes\
  -Wnested-externs\
  -Wall \
  #  -Wundef -Wwrite-strings -Wredundant-decls -Winline


srcdir = .
#VPATH = .:./../extras
LDFLAGS =  -g,-Wl,-T,/usr/local/arm-elf/lib/elf2flt.ld -elf2flt
LIBS =
CFLAGS = -g -O2 -pipe -Wall -I.

# Change these if necessary

YACC = bison -y
LEX = flex
CC = /usr/local/bin/arm-elf-gcc
CPP = /usr/local/bin/arm-elf-gcc -E
SOURCES = hello.c\

OBJS =  $(SOURCES:.c=.o)

all:  hello

hello: $(OBJS)
   $(CC) -o $@ $^ $(LDFLAGS) $(LIBS)

clean:
   rm -f $(OBJS) hello core *.gdb
```

ATTENTION

There is another example that involves modifying the Makefile. If the target source code file is **tcps1.c**, then users should modify at least 4 places to replace **hello** with **tcps1**.

```
srcdir = .

LDFLAGS =  -g,-W1,-T,/usr/local/arm-elf/lib/elf2flt.ld -elf2flt
LIBS =
CFLAGS = -g -02 -pipe -Wall -I.



YACC = bison -y
LEX = flex
CC = /usr/local/bin/arm-elf-gcc
CPP = /usr/local/bin/arm-elf-gcc -E
SOURCES = tcps1.c

OBJS = $(SOURCES:.c=.o)

all:   tcps1

tcps1: $(OBJS)
       $(CC) -o $@ $^ $(LDFLAGS) $(LIBS)

clean:
       rm -f $(OBJS) tcps1 core *.gdb
```

Copy the Hello example's Makefile to your source code target directory, and then modify it to work with your program. After that, follow the procedures outlined in the previous sections: *Compiling Hello.c*, *Uploading "Hello" to UC-7110*, and *Running "Hello" on UC-7110*.

3

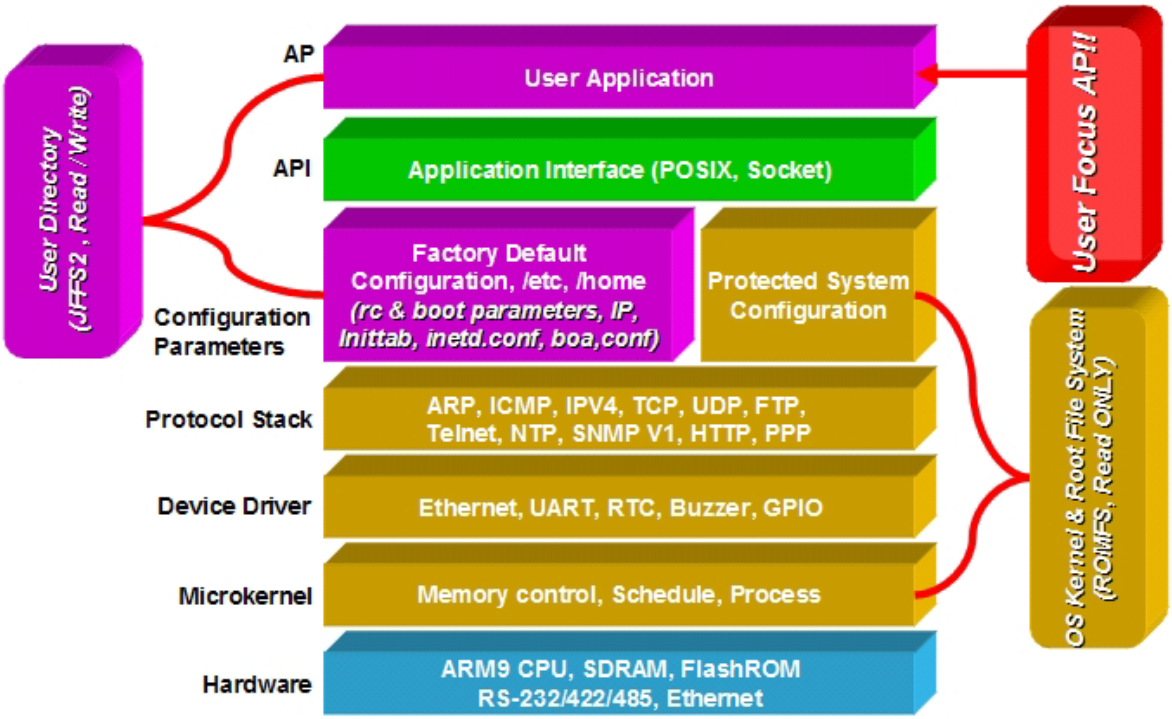
Software Package

This chapter includes information about the software that is used with UC-7110 Series products. This chapter covers the following topics:

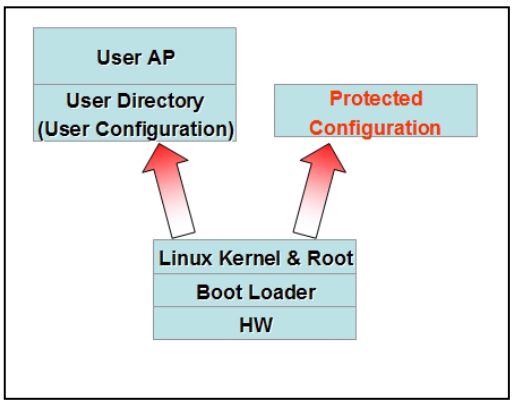
- **UC-7110 Software Architecture**
- **UC-7110 Software Package**

UC-7110 Software Architecture

The pre-installed µClinux Operating System used by UC-7110 follows the standard µClinux architecture, making programs that follow the POSIX standard easily ported to UC-7110 by using the GNU Tool Chain provided by www.uClinux.org. In addition to the Standard POSIX API, device drivers for the buzzer, and UART for the serial ports are also included.



UC-7110's Flash ROM has more than one smaller partition, which contain the **Boot Loader**, **Linux Kernel & Root (/) File System Image**, and **User Directory** partitions.



For most applications, users often spend a lot time maintaining the operating system and modifying the system configuration. In order to save on the total cost of development and maintenance, UC-7110 is specially design to partition a "User Directory" for storing the user's system configuration parameters.

UC-7110 has a built-in mechanism that prevents system crashes, helping preserve system reliability. The procedure is described below.

When the Linux kernel boots up, the kernel mounts the root file system and then enables services and daemons. The kernel also looks for the system configuration parameters via rc or inittab.

Normally, the kernel uses the User Directory to boot up the system. The kernel will only use the default configuration _etc & _home when the User Directory crashes.

UC-7110 uses ROMFS for the Linux kernel image, Root File System, and Protected configuration,

and uses JFFS2 for the User Directory.

The partition sizes are hard coded into the kernel binary. You must rebuild the kernel to change the partition sizes. The flash memory map is shown in the following table.

Flash Context	Flash Address	Size	Access control
Boot loader	0 – 0x3ffff	256 K	Read ONLY
Kernet & Root File System	0x40000– 0x3fffff	4 M - 256 K	Read ONLY **ROMFS**
User Directory	0x400000 – 0x7fffff	4 M	Read / Write **JFFS2**

In addition to the flash file systems, a RAM based file system will be mounted on /var/.

Read-Only File System (ROMFS)

ROMFS is a read-only file system for Linux. This means that it is a non-journaling file system. Unlike CRAMFS, ROMFS does not support code compression.

A working Linux system requires the kernel and at least some programs, so obviously a file system is also needed. Most Linux disk file systems are designed to be high performance, supporting all POSIX features, and sometimes elaborate recovery from crashes (journaling). This makes the system quite heavy-weighted, and thus often inappropriate for some special purposes.

Additional information about ROMFS is available at:

http://romfs.sourceforge.net/

Journaling Flash File System (JFFS2)

The flash User Directory is formatted by the **Journaling Flash File System (JFFS2)**, which places a compressed file system on the flash, transparent to the user.

The Journaling Flash File System (JFFS2) was developed by Axis Communications in Sweden.

JFFS2 provides a file system directly on flash, rather than emulating a block device designed for use on flash-ROM chips, and it recognizes flash-ROM chips' special write requirements, does wear-leveling to extend flash life, keeps the flash directory structure in RAM at all times, and implements a log-structured file system that is always consistent—even if the system crashes or unexpectedly powers down. It does not require fsck on boot up.

JFFS2, the next version of JFFS, provides improved wear-leveling and garbage-collection performance, improved RAM footprint and response to system-memory pressure, improved concurrency and support for suspending flash erases, marking of bad sectors with continued use of the remaining good sectors (thus enhancing the write-life of the devices), native data compression inside the file system design; and support for hard links.

Key features of JFFS2 are:

- Directly targeted to Flash ROM
- Robust
- Consistent across power failure
- No integrity scan (fsck) is required at boot time after normal or abnormal shutdown
- Explicit wear leveling
- Transparent compression

Although JFFS2 is a journaling file system, this does not preclude the loss of data. The file system will remain in a consistent state across power failures, and will always be mountable. However, if the board is powered down during a write, then the incomplete write will be rolled back on the next boot. Any writes that were already completed will not be affected.

Additional information about JFFS2 is available at:
http://sources.redhat.com/jffs2/jffs2.pdf
http://developer.axis.com/software/jffs/
http://www.linux-mtd.infradead.org/

UC-7110 Software Package

bin	dev
upkernel	mtdblock1
passwd -> tinylogin	mtdr1
login -> tinylogin	mtd1
tinylogin	mtdblock0
telnetd	mtdr0
snmpd	mtd0
mail	cum1
sh	cum0
routed	ttyM1
netstat	ttyM0
arp	urandom
chat	random
pppd	zero
portmap	ttypf
ntpdate	ttype
necid	ttypd
eraseall	ttypc
kversion	ttypb
init	ttypa
expand	ttyp9
inetd	ttyp8
hwclock	ttyp7
ftpd	ttyp6
ftp	ttyp5
mke2fs	ttyp4
e2fsck	ttyp3
discard	ttyp2
dhcpcd	ttyp1
cpu	ttyp0

bin	dev
busybox	ttyS0
boa	tty3
bf	tty2
backupfs	tty1
downramdisk	tty0
upramdisk	rom1
	rom0
	ptypf
	ptype
	ptypd
	ptypc
	ptypb
	ptypa
	ptyp9
	ptyp8
	ptyp7
	ptyp6
	ptyp5
	ptyp4
	ptyp3
	ptyp2
	ptyp1
	ptyp0
	ppp
	pio
	rtc
	ram1
	ram0
	null
	kmem
	mem
	cua0
	console
	tty

4

Configuring UC-7110

In this chapter, we describe how to configure UC-7110 Series products.

The following topics are covered in this chapter:

- **Enabling and Disabling Daemons**
- **Adding a Web Page**
- **Configuring Dial-in/Dial-out Service**
- **How to mount a remote NFS Server**
- **Upgrading the Kernel & Root File System**
- **Upgrading the User Directory**
- **User Directory Backup—UC-7110 to PC**
- **Loading Factory Defaults**
- **Mirroring the Application Program and Configuration**
- **How to Autostart User Applications on Bootup**
- **Checking the Kernel and User Directory Versions**

Enabling and Disabling Daemons

The following daemons are enabled when UC-7110 boots up for the first time.

- SNMP Agent daemon: **snmpd**
- Telnet Server / Client daemon: **telnetd**
- Internet Daemons: **inetd**
- FTP Server / Client daemon: **ftpd**
- WWW Server daemon: **boa**

ATTENTION

How to enable/disable telnet/ftp server

a. Edit the file '/etc/inetd.conf'
Example (default enable):
discard dgram udp wait root /bin/discard
discard stream tcp nowait root /bin/discard
telnet stream tcp nowait root /bin/telnetd
ftp stream tcp nowait root /bin/ftpd -l
b. Disable the daemon by typing '#' in front of the first character of the row.

How to enable/disable /etc/inittab www server

a. Edit the file '/etc/inittab'
b. Disable the www service by typing "#" in front of the first character of the row.

How to enable Network Time Protocol

ntpdate is a time adjusting client utility. UC-7110 plays the role of Time client, and sends requests to the Network Time Server to request the correct time.

Set the time server address for adjusting the system time with the command:

```
/>ntpdate ntp_server_ip
```

Save the system time to the hardware's real time clock, with the command:

```
/>hwclock --systohc
```

Visit http://www.ntp/org for a recommended public NTP server list.

How to update the system time periodically via Network Time Protocol

1. Create a shell script file that includes the following description.

```
#!/bin/sh
ntpdate ntp_server_ip
hwclock -systohc
sleep 100          ← The min time is 100ms.
```

2. Save and change this shell script to run-able by typing

```
chmod 755 <shell-script_name>
```

3. Edit the file '/etc/inittab' by adding the following line:

```
ntp: unkown: /directory/<shell_script_name>
```

Adding a Web Page

Default Home Page address:

/home/httpd/index.html

You may change the default home page directory by editing the web server's configuration file, located at: **/home/httpd/boa.conf**

```
/mnt/home/httpd> ls -al
-rw-r--r--    1 1000     100           333 index.html
-rw-r--r--    1 1000     100           437 boa.conf
drwxr-xr-x    4 0        0            1024 ..
drwx------    2 0        0            1024 .
/mnt/home/httpd> vi boa.conf
```

Type the following command to edit the boa.conf file:

/mnt/home/httpd>vi boa.conf

```
#
# A minimal config that makes the home page
# an unauthenticated CGI
#
ServerName uClinux
DocumentRoot /home/httpd
ScriptAlias /cgi-bin/ /home/httpd/cgi-bin/
Alias /img /home/httpd/img
# Auth /cgi-bin/cgi_demo /etc/config/config
AddType text/plain      txt
AddType image/gif       gif
AddType text/html       html
AddType text/html       htm
AddType text/xml        xml
AddType image/jpeg      jpe
AddType image/jpeg      jpeg
AddType image/jpeg      jpg
AddType image/x-icon    ico
~
~
```

To add your web page, place your home page in the following directory:

/home/httpd/

Configuring Dial-in/Dial-out Service

Dial-out Service

Direct cable connection:

- *Without* username and password, use:
  />pppd connect 'chat –v' /dev/ttyM0 38400 crtscts&
- *With* username and password, use:
  />pppd connect 'chat –v' user xxxxx password xxxxx /dev/ttyM0 38400 crtscts&

Connect Using a Modem:

- Use:
  />pppd connect 'chat –v ATDT<phone_number> CONNECT' user xxxxx password xxxxx /dev/ttyM0 38400 crtscts&

ATTENTION

If dial out fails, the pppd connection will be blocked, and the user will need to shut down the pppd, and re-dial. Since the return value is always OK (regardless of whether or not the connection is blocked), the AP must be set up to check the network status to determine if the connection is complete.

Dial-in Service

Direct cable connection:

- Use either of the following:
  />pppd <Local_IP_Address>:<Remote_IP_Address> /dev/ttyM1 38400 local crtscts
  or
  />pppd <Local_IP_Address>:<Remote_IP_Address> /dev/ttyM0 38400 local crtscts login auth

Connect Using a Modem:

- Use:
  />pppd connect 'chat –v AT CONNECT'
  <local_IP_Address>:<Remote_IP_Address> /dev/ttyM0 38400 crtscts login auth

How to Mount a Remote NFS Server

Currently, UC-7110 only supports NFS (Network File System) clients. Users can open NFS service on a Linux PC for UC-7110 to push data to. UC-7110 can use NFS to mount a remote disk as a local disk for data or log purposes.

1. First, the NFS server must open an export directory and allow access to the IP address. Edit the file "/etc/exports" on your Linux PC, and then run the NFS daemon. The following example gives one possibility (refer to the NFS-HOWTO document at http://nfs.sourceforge.net/nfs-howto/server.html):

   /home/usr 192.168.3.1 (rw,no_root_squash,no_all_squash)

2. UC-7110 must run the "portmap" utility. This program is enabled by default in the "/etc/rc" file. Use the following command to mount the remote NFS server:

   />mount –t nfs <remote-ip>:<remote-export-directory> <local-directory>

Upgrading the Kernel & Root File System

UC-7110 combines the kernel and root file system into the file ***uc7110-1.x.x.bin***, which can be downloaded from www.moxa.com. You must first download this file to your PC, and then use Console Terminal or Telnet Console to copy the file to UC-7110.

You can save this file to the UC-7110's RAM disk, and then upgrade the kernel and root file system. In what follows, we give a step by step example.

To enable the RAM disk, use the commands given below:

/>upramdisk

```
Command: #!/bin/sh
Command: mke2fs /dev/ram1
mke2fs 1.25 (20-Sep-2001)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
1152 inodes, 4608 blocks
230 blocks (4%) reserved for the super user
First data block=1
1 block group
8192 blocks per group, 8192 fragments per group
1152 inodes per group

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 25 mounts or
180.00 days, whichever comes first.  Use tune2fs -c or -i to override.
Command: mount /dev/ram1 /ramdisk
ext3: No journal on filesystem on ramdisk(1,1)
Command: rmdir /ramdisk/lost+found
Execution Finished, Exiting
```

/>cd ramdisk

Use the built-in FTP client to download the **uc7110-1.x.x.bin** file from the PC.

/ramdisk>ftp <destination PC's IP>

Login Name: xxxx

Login Password: xxxx

ftp> bin

ftp> get uc7110-1.x.x.bin

```
/> cd /ramdisk
ramdisk> ftp 192.168.3.11
Connected to 192.168.3.11.
220 TYPSoft FTP Server 1.10 ready...
Name (192.168.3.11:root): root
331 Password required for root.
Password:
230 User root logged in.
ftp> bin
200 Type set to I.
ftp> get uc7110-1.3.11.bin
local: uc7110-1.3.11.bin remote: uc7110-1.3.11.bin
200 Port command successful.
150 Opening data connection for uc7110-1.3.11.bin.
226 Transfer complete.
2481236 bytes received in 3 secs (682 Kbytes/sec)
ftp> quit
221 Goodbye!
/ramdisk> ls -al
-rw-r--r--    1 0        0         2481236 uc7110-1.3.11.bin
drwxr-xr-x    1 0        0              32 ..
drwxr-xr-x    2 0        0            1024 .
/ramdisk>
```

And then use the **upkernel** command to upgrade the kernel and root file system.

/ramdisk>upkernel uc7110-1.3.11.bin

/ramdisk>reboot

```
/ramdisk> ls -al
-rw-r--r--    1 0        0         2481236 uc7110-1.3.11.bin
drwxr-xr-x    1 0        0              32 ..
drwxr-xr-x    2 0        0            1024 .
/ramdisk> upkernel uc7110-1.3.11.bin
This step will destroy your old kernel.
Do you want to continue it ? (Y/N) : y
Format kernel and root file system disk !!!
Erased 3840 Kibyte @ 0 -- 100% complete.
Format OK. Now update the kernel and root file system.
Please wait ...
Update the kernel and root file system OK. Please restart system.
/ramdisk> reboot

The system is going down NOW !!
Sending SIGTERM to all processes.
```

Upgrading the User Directory

UC-7110 uses ROMFS (ROM File System), which is pre-set to READ only, for the kernel and root file system. UC-7110 provides a read/write user's directory in the JFFS2 file system. By using this user's directory, the system configuration file and user's program can be stored on this disk.

Search UC-7110's CD-ROM for the latest user directory file, or download the file from www.moxa.com. The format is **usrdisk-1.x.x.dsk**. You must download this file to a PC first, and then use Console Terminal or Telnet Console to copy the file to the UC-7110.

You can save this file to UC-7110's RAM disk, and then upgrade the user directory. In what follows, we give a step by step example.

To enable the RAM disk, use the commands given below:

/>upramdisk

```
Command: #!/bin/sh
Command: mke2fs /dev/ram1
mke2fs 1.25 (20-Sep-2001)
Filesystem label=
CS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
1152 inodes, 4608 blocks
230 blocks (4%) reserved for the super user
First data block=1
1 block group
8192 blocks per group, 8192 fragments per group
1152 inodes per group

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 25 mounts or
180.00 days, whichever comes first.  Use tune2fs -c or -i to override.
Command: mount /dev/ram1 /ramdisk
ext3: No journal on filesystem on ramdisk(1,1)
Command: rmdir /ramdisk/lost+found
Execution Finished, Exiting
```

/>cd ramdisk

Use the built-in FTP client to download the **usrdisk-1.x.x.dsk** file from the PC.

/ramdisk>ftp <destination PC's IP>

Login Name: xxxx

Login Password: xxxx

ftp> bin

ftp> get usrdisk-1.x.x.dsk

ftp>quit

/ramdisk>bf /ramdisk/usrdisk-1.x.x.dsk

/reboot

```
ftp> get usrdisk-1.3.11
local: usrdisk-1.3.11 remote: usrdisk-1.3.11
200 Port command successful.
150 Opening data connection for usrdisk-1.3.11.
WARNING! 705 bare linefeeds received in ASCII mode
File may not have transferred correctly.
226 Transfer complete.
4194336 bytes received in 38 secs (107 Kbytes/sec)
ftp> quit
221 Goodbye!
/ramdisk> ls -al
-rw-r--r--    1 0        0         4194308 usrdisk-1.3.11
drwxr-xr-x    1 0        0              32 ..
drwxr-xr-x    2 0        0            1024 .
/ramdisk> bf /ramdisk/usrdisk-1.3.11
This setp will destroy your user directory (/mnt) all data.
Do you backup it and continue to do it ? (Y/N) : y
Umount user directory.
umount: /dev/mtdblock1 busy - remounted read-only
Format user directory !!!
Erased 4096 Kibyte @ 0 -- 100% complete.
Format OK. Now make the user directory.
Please wait ...
Make user directory OK. Please restart system.
/ramdisk>
```

User Directory Backup—UC-7110 to PC

To enable the RAM disk, follow the commands given below:

/>upramdisk

/>cd ramdisk

And then use the **backupfs** command to backup the file system.

/ramdisk>backupfs /ramdisk/usrdisk-backup

```
/> backupfs /ramdisk/usrdisk-backup
Sync the file system.
Now backup the user directory. Please wait ...
Backup the user directory OK.
/>
```

The file system will be backed up, and you can use ftp commands to transfer the **usrdisk-backup** to the FTP server on the PC.

```
/> cd /ramdisk
/ramdisk> ls -al
----------    1 0        0         4194304 usrdisk-backup
drwxr-xr-x    1 0        0              32 ..
drwxr-xr-x    2 0        0            1024 .
/ramdisk> ftp 192.168.3.11
Connected to 192.168.3.11.
220 TYPSoft FTP Server 1.10 ready...
Name (192.168.3.11:root): root
331 Password required for root.
Password:
230 User root logged in.
+bin
ftp> put usrdisk-backup
local: usrdisk-backup remote: usrdisk-backup
200 Port command successful.
150 Opening data connection for usrdisk-backup.
226 File received complete
4195224 bytes sent in 37 secs (113 Kbytes/sec)
ftp>
```

Loading Factory Defaults

The easiest way to **"Load Factory Default"** is with "Upgrade User directory."

Refer to the previous section **"How to Upgrade User Directory"** for an introduction.

Mirroring the Application Program and Configuration

For some applications, you may need to "Mirror" (or sometimes we said "Ghost") one UC-7110's user directory, and duplicate it to other UC-7110s.

To do this, use the following recommended procedure:

1. Backup the user directory to a PC.
   (Refer to the previous topic **User Directory Backup—UC-7110 to a PC**.)

   Hint:

   /ramdisk>backupfs /ramdisk/<user defined file name>

2. Download the backed up user directory to the other UC-7110.
   (Refer to the previous topic **Upgrading the User Directory**)

   Hint:

   /ramdisk>bf /ramdisk/<User directory file name>

How to Autostart User Applications on Bootup

Edit the **/etc/rc** file by adding your application program. E.g.,

/ap-directory/ap-program &

Checking the Kernel and User Directory Versions

Use the following commands to check the version of the kernel and root file system:

To check the kernel version:

/>kversion

To check the user directory version:

/>cat /etc/version

5

UC-7110 Device API

In this chapter, we discuss the Device API for the UC-7110 Series. We introduce the APIs for the following functions:

- **RTC (Real Time Clock)**
- **Buzzer**
- **UART Interface**

RTC (Real Time Clock)

The device node is located at **/dev/rtc**. UC-7110 supports µClinux standard simple RTC control. You must include **<linux/rtc.h>** to use these functions.

1. Function: RTC_RD_TIME

```
int ioctl(fd, RTC_RD_TIME, struct rtc_time *time);
```

Description: Reads time information from RTC.

2. Function: RTC_SET_TIME

```
int ioctl(fd, RTC_SET_TIME, struct rtc_time *time);
```

Description: Sets RTC time.

Buzzer

The device node is located at **/dev/console**. UC-7110 supports µClinux standard buzzer control. UC-7110's buzzer runs at a fixed frequency of 100 Hz. You must include **<sys/kd.h>** to use these functions.

1. Function: KDMKTONE

```
ioctl(fd, KDMKTONE, unsigned int arg);
```

Description: Buzzer will beep, as stipulated by the function arguments.

UART Interface

The normal tty device node is located at **/dev/ttyM0…ttyM1**, and modem tty device node is located at **/dev/cum0 … cum1**. UC-7110 Series supports µClinux standard termios control. Moxa UART Device API supports the configuration of ttyM0 to ttyM1 as RS-232/422/485. You must include **<moxadevice.h>** to use these functions.

```
#define RS232_MODE          0
#define RS485_2WIRE_MODE        1
#define RS422_MODE          2
#define RS485_4WIRE_MODE        3
```

1. Function: MOXA_SET_OP_MODE

```
int mode;
mode=which mode you want to set;
int ioctl(fd, MOXA_SET_OP_MODE, &mode)
```

Description: Sets the interface mode.

2. Function: MOXA_GET_OP_MODE

```
int mode;
int ioctl(fd, MOXA_GET_OP_MODE, &mode)
```

Description: Gets the interface mode.

6

UC Finder

UC-7110 comes with a UC Finder utility, which has the sole purpose of searching the LAN or intranet for UC-7110 units.

For most of applications, it's not easy to remember the IP addresses of Universal Communicators connected to the LAN. This is especially true for some problem solving and testing in the field. The UC Finder utility broadcasts messages over the LAN to search for IP addresses of Universal Communicators connected to the LAN. UC Finder searches for the class of MAC addresses assigned to Universal Communicators. UC-7110 supports the GUI-style Windows UC Finder, and also supports a command line utility for Linux environments.

In this chapter, we discuss the following UC Finder topics:

- **Windows UC Finder**
- **Linux UCfinder**

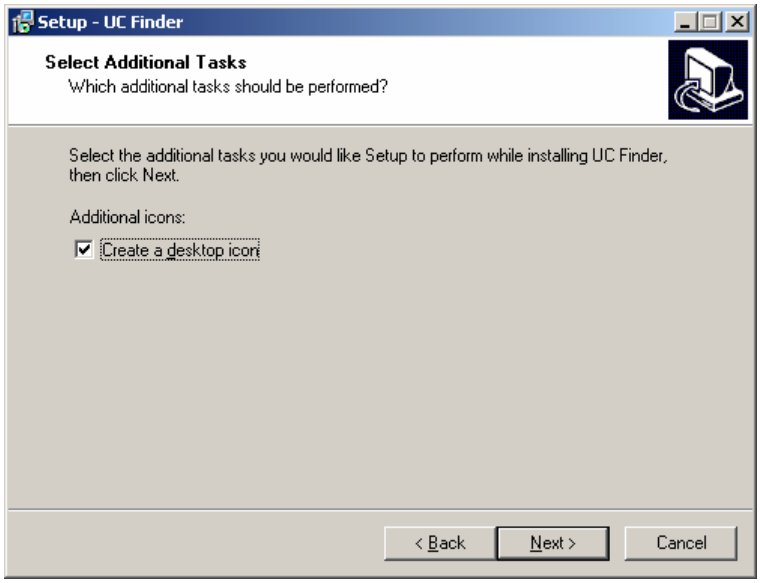
Windows UC Finder

The following steps describe how to install UC Finder on a Windows PC.

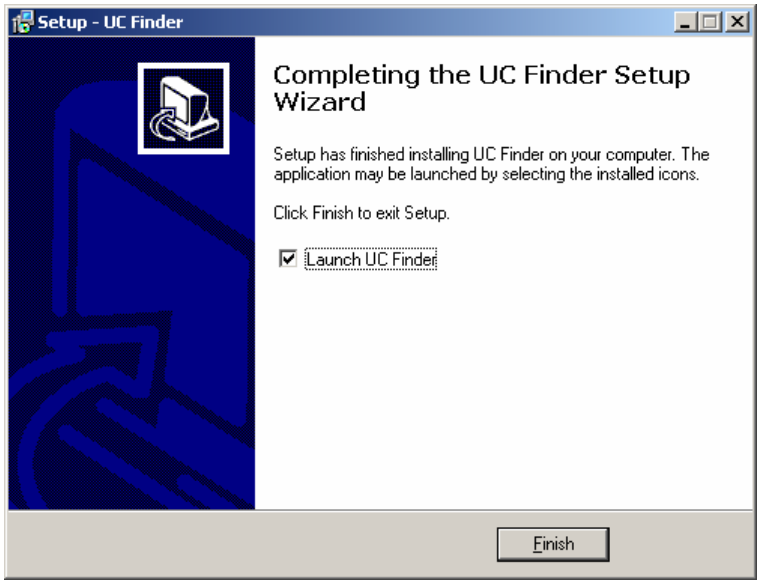
1. Double click on the UC Finder installation program, **Setup.exe**, to start the installation.
2. When the **Welcome to the UC Finder Setup Wizard** window opens, click on **Next** to continue.

Setup - UC Finder
Welcome to the UC Finder Setup Wizard
This will install UC Finder on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
Next >
Cancel

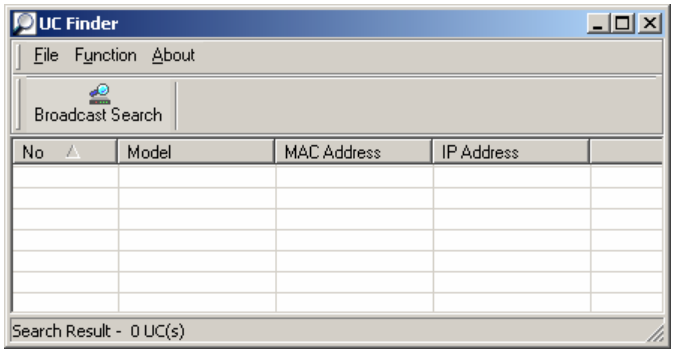
3. Check the **Create a desktop icon** box, and then click on **Next** to continue.


Setup - UC Finder
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing UC Finder, then click Next.
Additional icons:
Create a desktop icon
< Back
Next >
Cancel

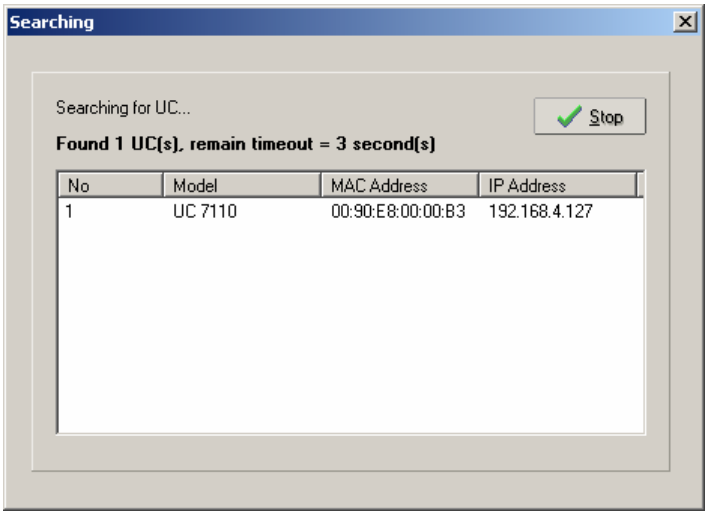
4. Check the **Launch UC Finder** checkbox to use UC Finder immediately after the installation has finished, and then click on **Next** to complete the installation.


Setup - UC Finder
Completing the UC Finder Setup Wizard
Setup has finished installing UC Finder on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
Launch UC Finder
Finish

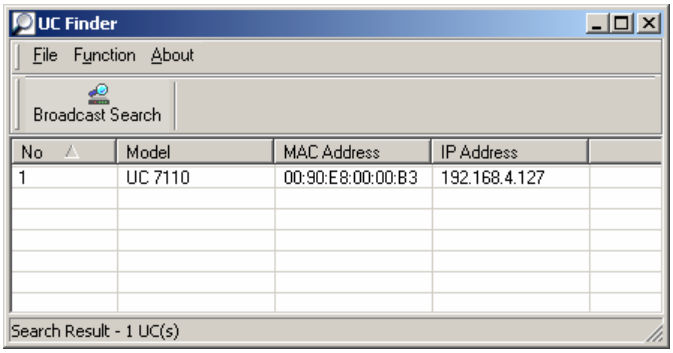
5. When the UC Finder window opens, click on **Broadcast Search** to search for all Universal Communicators connected to the LAN.



6. The **Searching** window will show the Universal Communicators that have been located. You can click on **Stop** as soon as the Universal Communicator you are looking for is listed.



7. When the search is complete, the Broadcast Search window closes, and the **Model**, **MAC Address**, and **IP Address** of all Universal Communicators that were located will be listed in the UC Finder window.



ATTENTION

UC finder is designed solely to find IP addresses of networked Universal Communicators. It cannot be used to configure Universal Communicators over the network. If you need to configure UC's IP address or other parameters, connect to UC's console utility by Telnet (over the network) or serial console (using the serial console cable that came with the product).

The next time you need to run UC Finder, double click on the UC Finder icon located on your PC's desktop to launch this utility.

Linux ucfinder

To use the Linux **ucfinder** utility, copy ucfinder from the CD-ROM to your Linux PC, and then use the command given below to start ucfinder. The ucfinder utility will automatically broadcast a message over your LAN network to find the IP address of all UC's connected to the LAN.

#./ucfinder

A

System Commands

busybox (V0.60.4): µClinux normal command utility collection

File manager

cp	copy file
ls	list file
ln	make symbolic link file
mount	mount and check file system
rm	delete file
chmod	change file owner & group & user
chown	change file owner
chgrp	change file group
sync	sync file system; save system file buffer to hardware
mv	move file
pwd	display active file directly
df	list active file system space
mkdir	make new directory
rmdir	delete directory

Editor

vi	text editor
cat	dump file context
zcat	Compress or expand files
grep	print lines matching a pattern
cut	remove sections from each line of files
find	search for files in a directory hierarchy
more	dump file by one page
test	test if file exists or not
sleep	sleep (seconds)
echo	echo string

Network

ping	ping to test network
route	routing table manager
netstat	display network status
ifconfig	set network IP address
tracerout	trace route
tftp	tftp protocol
telnet	user interface to TELNET protocol
ftp	file transfer protocol

Process

kill	kill process
killall	kill process by name
ps	report process status

Other

dmesg	dump kernel log message
stty	set serial port
zcat	dump .gz file context
mknod	make device node
free	display system memory usage
date	print or set the system date and time
env	run a program in a modified environment
clear	clear the terminal screen
reboot	reboot / power off/on the server
halt	halt the server
du	estimate file space usage
gzip, gunzip	compress or expand files
hostname	show system's host name
tar	tar archiving utility

MOXA Special Utilities

backupfs	backup file system (user directory)
bf	build file system (user directory)
cat /etc/version	show user directory version
upramdisk	mount ramdisk
downramdisk	unmount ramdisk
kversion	show kernel version

B

SNMP Agent with MIB II & RS-232 Like Group

UC-7110 has a built-in SNMP (Simple Network Management Protocol) agent that supports RFC1317 RS-232 like group and RFC 1213 MIB-II. The following table lists the variable implementation for UC-7110.

Note: UC-7110 does not support SNMP trap.

RFC1213 MIB-II supported SNMP variables:

system MIB	interface MIB	at MIB	icmp MIB
sysDescr	ifNumber	atTable	icmpInMsgs
sysObjectID	ifTable	atIfIndex	icmpInErrors
sysUpTime	ifIndex		icmpInDestUnreachs
sysContact	ifDescr	atPhysAddress	icmpInTimeExcds
sysName	ifType		icmpInParmProbs
sysLocation	ifMtu	atNetAddress	icmpInSrcQuenchs
sysServices	ifSpeed		icmpInRedirects
	ifPhysAddress		icmpInEchos
	ifAdminStatus		icmpInEchoReps
	ifOperStatus		icmpInTimestamps
	ifLastChange		icmpInAddrMasks
	ifInOctets		icmpInAddrMaskReps
	ifInUcastPkts		icmpOutMsgs
	ifInNUcastPkts		icmpOutErrors
	ifInDiscards		icmpOutDestUnreachs
	ifInErrors		icmpOutTimeExcds
	ifInUnknownProtos		icmpOutParmProbs
	ifOutOctets		icmpOutSrcQuenchs
	ifOutUcastPkts		icmpOutRedirects
	ifOutNUcastPkts		icmpOutEchos
	ifOutDiscards		icmpOutEchoReps
	ifOutErrors		icmpOutTimestamps
	ifOutQLen		icmpOutAddrMasks
	ifSpecific		icmpOutAddrmaskReps

ip MIB	tcp MIB	udp MIB
ipForwarding	tcpRtoAlgorithm	udpInDatagrams
ipDefaultTTL	tcpRtoMin	udpNoPorts
ipInReceives	tcpRtoMax	udpInErrors
ipInHdrErrors	tcpMaxConn	udpOutDatagrams
ipInAddrErrors	tcpActiveOpens	udpTable
ipForwDatagrams	tcpPassiveOpens	udpLocalAddress
ipInUnknownProtos	tcpAttemptFails	udpLocalPort
ipInDiscards	tcpEstabResets	
ipInDelivers	tcpCurrEstab	
ipOutRequests	tcpInSegs	
ipOutDiscards	tcpOutSegs	
ipOutNoRoutes	tcpRetransSegs	
ipReasmTimeout	tcpConnTable	
ipReasmReqds	tcpConnState	
ipReasmFails	tcpConnLocalAddress	
ipFragOKs	tcpConnLocalPort	
ipFragFails	tcpConnRemAddress	
ipFragCreates	tcpConnRemPort	
ipAddrTable	tcpInErrs	
ipAdEntAddr	tcpOutRsts	
ipAdEntIfIndex		
ipAdEntNetMask		
ipAdEntBcastAddr		
ipAdEntReasmMaxSize		
ipRouteTable		
ipRouteDest		
ipRouteIfIndex		
ipRouteMetric1		
ipRouteMetric2		
ipRouteMetric3		
ipRouteMetric4		
ipRouteNextHop		
ipRouteType		
ipRouteProto		
ipRouteAge		
ipRouteMask		
ipRouteMetric5		
ipRouteInfo		
ipNetToMediaTable		
ipNetToMediaIfIndex		
ipNetToMediaPhysAddress		
ipNetToMediaNetAddress		
ipNetToMediaType		
ipRoutingDiscards		

snmp MIB
snmpInPkts
snmpOutPkts
snmpInBadVersions
snmpInBadCommunityNames
snmpInBadCommunityUses
snmpInASNParseErrs
snmpInTooBigs
snmpInNoSuchNames
snmpInBadValues
snmpInReadOnlys
snmpInGenErrs
snmpInTotalReqVars
snmpInTotalSetVars
snmpInGetRequests
snmpInGetNexts
snmpInSetRequests
snmpInGetResponses
snmpInTraps
snmpOutTooBigs
snmpOutNoSuchNames
snmpOutBadValues
snmpOutGenErrs
snmpOutGetRequests
snmpOutGetNexts
snmpOutSetRequests
snmpOutTraps
snmpEnableAuthenTraps

RFC1317 RS-232 like group supported variables

rs232 MIB
rs232Number
rs232PortTable
rs232PortIndex
rs232PortType
rs232PortInSigNumber
rs232PortOutSigNumber
rs232PortInSpeed
rs232PortOutSpeed
rs232AsyncPortTable
rs232AsyncPortIndex
rs232AsyncPortBits
rs232AsyncPortStopBits
rs232AsyncPortParity
rs232InSigTable
rs232InSigPortIndex
rs232InSigName
rs232InSigState
rs232OutSigTable
rs232OutSigPortIndex
rs232OutSigName
rs232OutSigState

C

FAQ for UC-7110

FAQ 1 Why can I only use vfork( ),and am unable to use fork( )?

Answer 1 uClinux only supports vfork( ). It does not support fork( ). Note that when using vfork( ), the parent process will hang until the son process calls an exec group API, or exits.

FAQ 2 When using a pthread group API, why can't I use SIGUSR1 and SIGUSR2?

Answer 2 Since a pthread group API uses SIGUSR1 and SIGUSR2 to do a pthread control suspend, restart exit function, we cannot use the SIGUSR1 and SIGUSR2 signals. You will get the same result if you link the pthread. This means that you cannot use -1pthred to add an option to the linker.

FAQ 3 What is the correct format for linking to an AP?

Answer 3 **`arm-elf-gcc –W1, -elf2flt`**
(In this example, the AP converts elf format to flat format.)

FAQ 4 What stack size can I use when creating a user application?

Answer 4 First note that UC-7110 is not an MMU processor. The OS is uClinux, so you must transfer your application from ELF to FLAT format. The transfer utility (on the PC) assigns a stack size of 0x1000 by default. If your application needs a bigger stack size, you must use an option that specifies stack size when transferring the application from ELF to FLAT format. The following LDFLAGS shows how to write the stack size option:

```
LDFLAGS = -g,-Wl,-T,/usr/local/arm-elf/lib/elf2flt.ld
-elf2flt= "-s 32768"
```

You can combine this with a compress option. E.g.:

```
LDFLAGS = -g,-Wl,-T,/usr/local/arm-elf/lib/elf2flt.ld
-elf2flt= "-z –r -s 32768"
```

Refer to the next FAQ for compress options.

Keep in mind that the stack size must be indicated in decimal format. You cannot use hex format (i.e., numbers that begin with "0x"). The following LDFLAGS command will fail:

```
LDFLAGS = -g,-Wl,-T,/usr/local/arm-elf/lib/elf2flt.ld
-elf2flt= "-s 0x2000"
```

Use the 'flthdr' utility (on a PC) to check the current stack size.

FAQ 5 How do I compress an application program?

Answer 5 UC7110's kernel supports ZFLAT format files. This means that when you compile the program, you can add the compress option to reduce the size of the binary file. Doing so will reduce the required flash space on the user disk. The disadvantage of using the compress option is that it takes about 5% to 10% more time when loading the program from Flash to RAM. After uncompressing and loading the program to RAM, there is no difference. You just need to modify the makefile, LDFLAGS, as follows:

LDFLAGS = -g,-WI,-T,/usr/local/arm-elf/lib/elf2flt.ld -elf2flt= "-z –r"

NOTE: The user directory's JFFS2 file system also has a compress function. If you compile the program with the ZFLAT feature, the file will only be compressed by an additional 5% to 10% when you copy the program into JFFS2 Flash.

FAQ 6 Since UC-7110's tool chain currently does not support GDB server, how do I disable a function by using the Makefile example to generate the gdb file?

Answer 6 Edit the file **/usr/local/arm-elf/bin/ld** on the PC Linux platform. Add one line after line 133 (the content of line 133 is: **rm –f $OFILE.elf #not needed for any reason**) as follows: **rm –f $OFILE.gdb**. I.e., after you edit the file lines 133 and 134 will look like:

Line 133: **rm –f $OFILE.elf #not needed for any reason**
Line 134: **rm –f $OFILE.gdb**

D

Service Information

This appendix shows you how to contact Moxa for information about this and other products, and how to report problems.

In this appendix, we cover the following topics.

- **MOXA Internet Services**
- **Problem Report Form**
- **Product Return Procedure**

MOXA Internet Services

Customer satisfaction is our number one concern, and to ensure that customers receive the full benefit of our products, Moxa Internet Services has been set up to provide technical support, driver updates, product information, and user's manual updates.

The following services are provided

E-mail for technical support.....support@moxa.com.tw

World Wide Web (WWW) Site for product information:

.....http://www.moxa.com

Problem Report Form

MOXA UC-7110 Series

Customer name:	
Company:	
Tel:	**Fax:**
Email:	**Date:**

1. **Moxa Product:** ☐ UC-7110
2. **Serial Number:** _____

Problem Description: Please describe the symptoms of the problem as clearly as possible, including any error messages you see. A clearly written description of the problem will allow us to reproduce the symptoms, and expedite the repair of your product.

Product Return Procedure

For product repair, exchange, or refund, the customer must:

- Provide evidence of original purchase.
- Obtain a Product Return Agreement (PRA) from the sales representative or dealer.
- Fill out the Problem Report Form (PRF). Include as much detail as possible for a shorter product repair time.
- Carefully pack the product in an anti-static package, and send it, pre-paid, to the dealer. The PRA should be visible on the outside of the package, and include a description of the problem, along with the return address and telephone number of a technical contact.